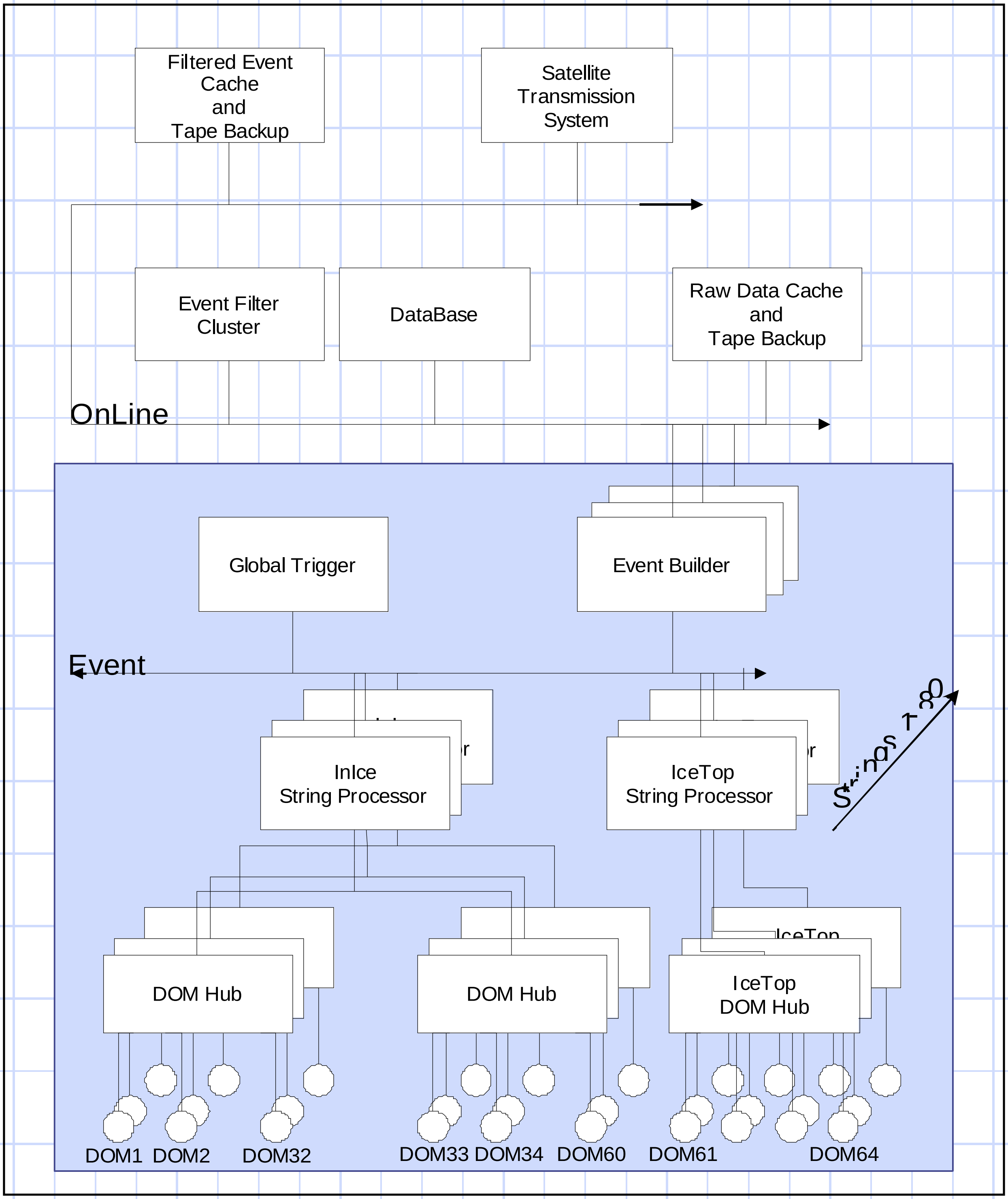
IceCube DAQ:
Implementation Plan



Filtered Event Cache and Tape Backup

Satellite Transmission System

Event Filter Cluster

DataBase

Raw Data Cache and Tape Backup

OnLine

Global Trigger

Event Builder

Event

InIce String Processor

IceTop String Processor

Strings 1-80

DOM Hub

DOM Hub

IceTop DOM Hub

DOM1 DOM2 DOM32

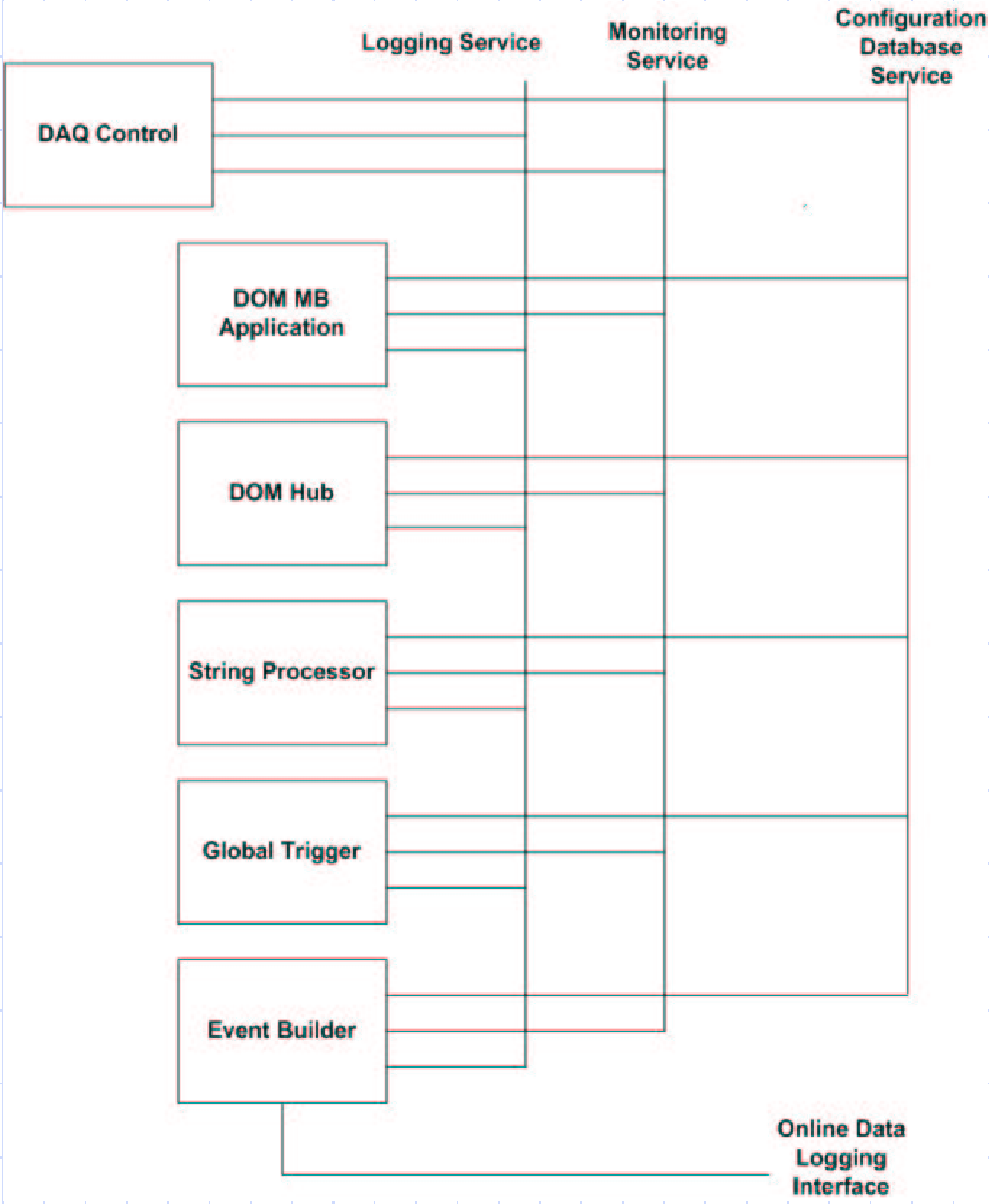
DOM33 DOM34 DOM60

DOM61 DOM64

Implementation Strategy:

- Parallel development Impt.
- Use of simulated subsystems.
  - Simulated DOM bootstrap and application allows DOM Hub development without DOM or DCI hardware.
  - Early verification of system correctness for large numbers of DOMs and Hubs.
- Use of simulated data.
  - Injected data allows testing for system “correctness”.
  - Injected data allows efficiency testing and validation of global trigger.
- Target what’s needed when.
  - Interfaces are important. Implementations can evolve.

DAQ Design Components:



DOM Main Board Application:

- Minimum rewrite of existing String 18 application.
- Use real and simulated access library to isolate program from hardware.
  - Bootstrap tests w/o DOM.
  - Slow control tests w/o DOM.
- Injection of simulation data will allow testing of full string DOM data streams before string is available.
- What/when
  - Low level tests are needed first (more on Tues.)
  - DOM application is needed for any real data analysis.
  - DOM feature extraction and/or compression is needed for full data rate testing or acquisition.

DOM Hub:

String Processor:

- Simple SP can be developed and used early, full SP will follow.
  - Hit synopsis not needed until we have multiple strings.
- Once DOM Hub output is defined, detailed design and impl. Can begin.
  - Initial data fmt. there.
  - DOM monitor and time correlation soon.
- Since SP performs time correction, needed for any multiDOM correlations.
  - Single string tests and external trigger tests.
  - Synopsis useful as early diagnostic.

Global Trigger:

- SP hit synopsis and GT core should proceed independently of surrounding infrastructure.
- Simple SP, as used for single string, is useful for DAQ control design and testing.
- Co-ordination with IceTop trigger model.
- Probably most productive area for use of simulation data.
  - Trigger condition validation.
  - Interplay of multiple trigger types on event builder behavior.
  - Trigger efficiency for all event types.
  - Derive trigger performance metrics for use with real data.

Event Builder:

- Once designed, should remain relatively independent of physics requirements.
- Initial single string event builder can simply provide flat file output.
- Simple shell will allow DAQ control development.
- Co-ordination of event builder data schema with IceTop data model.

Support Subsystems:

- Configuration DB, logging and monitoring are common to both DAQ and online.
  - DAQ needs some form of them first.
  - Overall system will benefit from common design and implementation.
  - Common user interface needed across DAQ and online
  - Design interface and use “OTS” tools initially (e.g. log4j).
- Online logging interface can be simple shell until online becomes populated.

Implementation issues:

- Investigation of available network bandwidth and messaging rates available @pole.
  - Test suite that can measure available rates for a given network/processor config. And able to verify perfromance during installation.
- Detailed description of real DAQ operational modes.
  - Calibration runs.
  - Expected operation during deployment and string integration.
- Requirements and design of unified experiment control supervisor.
  - What DAQ states should be supported?
- How do we inject physics content into global trigger framework?